# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-234263
(43)Date of publication of application : 27.08.1999

(51)Int.Cl. H04L 9/32
G09C 1/00
H04L 9/10

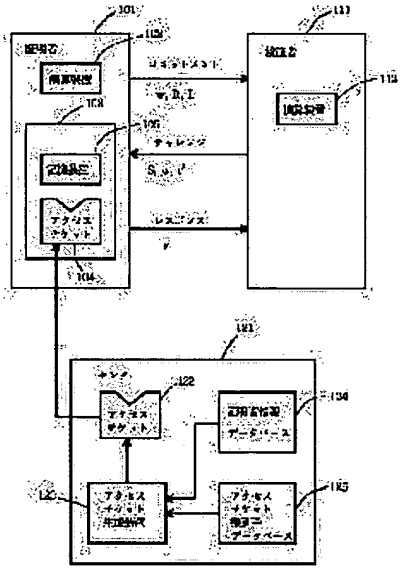
(21)Application number : 10-030053
(22)Date of filing : 12.02.1998

(71)Applicant : FUJI XEROX CO LTD
(72)Inventor : SUZUKI KOJI
NAKAGAKI JUHEI
SHIN YOSHIHIRO



## (54) METHOD AND DEVICE FOR MUTUAL AUTHENTICATION

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a mutual authentication suitable for a device of low arithmetic capacity.
SOLUTION: A testifier 101 generates a commitment (w) and sends it to a verifier 111. The verifier 111 sends challenges S, $\mu$ and r' to the testifier 101 and calculates an index g=h(r', $\mu$) for verification while using a unidirectional hash function (h) shared with the testifier 101. The testifier 101 verifies S while using disclosed information, further generates an index g'=h(r', $\mu$) for verification while using r', $\mu$ and (h), generates response generation information y'≤rF(d, L, It)g' mod n while using a unidirectional hash function F shared with a center 121, further generates a response y≡y'tg' mod n while using the index g' for verification and sends it to the verifier 111. The verifier 111 generates verification information w'≡JAgEyv mod n while using disclosed testifier information JA or the like of the testifier 101 and compares it with the commitment (w).



LEGAL STATUS

[Date of request for examination] 20.09.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

【特許請求の範囲】

【請求項１】　証明者が有資格者であることを保証するセンターが生成する有資格情報を用い、証明者側の有資格性、および検証者側の認証の双方をゼロ知識証明方式を用いて行う相互認証方法であって、証明者が不正な操作を施すことが困難な演算方法によって生成されたレスポンス生成情報に対して、センターが発行する有資格者情報を用いて該レスポンス生成情報に演算を施し、演算を施した情報をレスポンスとして検証者に提示することにより有資格者であることを証明することを特徴とする相互認証方法。

【請求項２】　証明者側の有資格性の証明を行うためのゼロ知識証明方式として、零化域を求めることが計算量的に困難な有限アーベル群を用いたゼロ知識証明方式を用いる請求項１記載の相互認証方法。

【請求項３】　検証者側の認証を行うためのゼロ知識証明方式として、零化域を求めることが計算量的に困難な有限アーベル群を用いたゼロ知識証明方式を用いる請求項１または２記載の相互認証方法。

【請求項４】　検証者側の認証を行うためのゼロ知識証明方式において、検証者の署名を用いた認証方法を用いる請求項１、２または３記載の相互認証方法。

【請求項５】　認証者側の認証を行うためのゼロ知識証明方式として、離散対数問題が計算量的に困難な有限群を用いたゼロ知識証明方式を用いる請求項１、２、３または４記載の相互認証方法。

【請求項６】　検証者側の認証を行うためのゼロ知識証明方式として、離散対数問題が計算量的に困難な有限群を用いたゼロ知識証明方式を用いる請求項１、２、３、４または５記載の相互認証方法。

【請求項７】　検証者から送られてくるチャレンジと共に認証者に対する指示情報を送信する請求項１、２、３、４、５または６記載の相互認証方法。

【請求項８】　証明器、検証器および有資格情報生成器を有し、証明者が有資格者であることを保証する有資格情報を用い、証明者側の有資格性、および検証者側の認証の双方をゼロ知識証明方式を用いて行う相互認証装置でにおいて、

上記有資格情報生成器は、上記証明者の有資格情報を生成する手段を有し、

上記証明器は、不正な操作を施すことが困難な演算方法によってレスポンス生成情報を生成する手段と、

上記レスポンス生成情報に対して、上記有資格情報生成器が発行した有資格者情報を用いて所定の演算を施しレスポンスを生成する手段とを有し、

上記検証器は、上記レスポンスを用いて検証を行なう検証手段を有することを特徴とする相互認証装置。

【請求項９】　上記証明器は証明者の演算装置、および該演算装置に接続された、外部から観察困難な装置内に保持された、演算装置及び記憶装置からなり、上記レスポンス生成情報の生成に不可欠な証明者に付帯した秘密情報を、該観察困難な装置内の記憶装置に保持し、該秘密情報を用いたレスポンス生成情報の生成を該外部から観察困難な装置内の演算装置において生成する請求項８記載の相互認証装置。

【請求項１０】　該外部から観察困難な装置内の演算装置をスマートカードでもって実現する請求項９記載の相互認証装置。

【請求項１１】　レスポンス生成情報からレスポンスを生成する演算を行うソフトウェアに対して難読化処理を施すことにより、不正な操作を行うことが困難にする請求項８、９または１０記載の相互認証装置。

【請求項１２】　検証者側の演算を行うソフトウェアに対して難読化処理を施すことにより、不正な操作を行うことが困難にする請求項８、９、１０または１１記載の相互認証装置。

【請求項１３】　外部から観測困難な装置内の情報を、検証者からの指示情報に基づいて書き換える請求項８、９、１０、１１または１２記載の相互認証装置。

【請求項１４】　上記外部から観測困難な装置内の情報として課金情報を含む請求項１３記載の相互認証装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、個人認証の分野に属し、特に検証者の認証も同時に行う相互認証の分野に属する。

【０００２】

【従来の技術】従来、認証を行う際にはＲＳＡ（Ｒｉｖｅｓｔ－Ｓｈａｍｉｒ－Ａｄｌｅｍａｎ）暗号を基本とした認証方式を用いるのが通常であった。ＲＳＡ暗号は、素因数分解の困難性にその安全性の根拠を置いた暗号方式で、以下の式を満たすように構成された公開鍵（ｅ，ｎ）、及び秘密鍵ｄを用いる。

【０００３】

【数１】$ed \equiv 1 \mod \phi(n)$

ただし、上でｎは二つ以上の素因数分解が困難なくらい大きな素数の積からなる合成数法数、φ（ｎ）は、乗法群（Ｚ＝ｎＺ）*の位数である。

【０００４】ＲＳＡ暗号を用いて認証を行う場合は、上記の秘密鍵で検証側から送信されるてくるチャレンジｃを暗号化して、レスポンスｒを生成する。つまり、証明側は

【０００５】

【数２】$r \equiv c^d \mod n$を計算して、検証側に送り返す。次てきたレスポンスｒに対して以下の計算を行い、ｒ’を求める。

【０００６】

【数３】$r' \equiv r^e \mod n$証明側が公開鍵（ｅ，ｎ）に対応

いるならば、チャレンジｃとｒ’は一致する。なぜならば、

【０００７】

【数４】ｅｄ≡１　ｍｏｄ　φ（ｎ）

より、

【０００８】

【数５】ｅｄ＝Ｑφ（ｎ）＋１

が成り立つ。ただし、上記でＱは適当な整数である。このときオイラーの定理より、

【０００９】

【数６】

$$\begin{aligned} r' &\equiv r^{e} \mod n \\ &\equiv c^{de} \mod n \\ &\equiv c^{Q\phi(n)+1} \mod n \\ &\equiv c\,(c^{Q})^{\phi(n)} \mod n \\ &\equiv c \mod n \end{aligned}$$

となるからである。チャレンジｃとｒ’が一致するとき、認証は成功したと見なされる。

【００１０】ＲＳＡ暗号を用いた相互認証を行う場合には、証明側の公開鍵（ｅ，ｎ）および秘密鍵ｄ、そして検証側の認証を行うための公開鍵（ｅ’，ｎ’）および秘密鍵ｄ’を用意する。（ｅ’，ｎ’）およびｄ’の生成条件は上記に述べた（ｅ，ｎ）およびｄの生成条件と同じである。そしてこれら二つの鍵ペアを用いて、証明側、検証側の双方の認証を行う。

【００１１】

【発明が解決しようとする課題】ＲＳＡ暗号では、秘密鍵ｄを法数ｎ程度の大きな数としなければならない。これは、秘密鍵ｄを小さい数とすると総当たり攻撃により、ｄの値を知られてしまうためである。

【００１２】ところで、べき乗剰余計算ｃｄ　ｍｏｄ　ｎの計算量は一般に指数ｄの値に比例する。通常、安全性のために法数ｎは１０２４ビット程度の非常に大きな数とされるため、ＲＳＡ暗号を用いた認証では、証明を行う側の計算量は膨大なものとなってしまう。

【００１３】個人認証などにおいて、証明側はスマートカード（ＩＣカード）などの演算能力の低いデバイスを用いることが多い。Ｂｒｕｃｅ　Ｓｃｈｎｅｉｎｅｒ，Ａｐｐｌｉｅｄ　Ｃｒｙｐｔｏｇｒａｐｈｙ（Ｓｅｃｏｎｄ　Ｅｄｉｔｉｏｎ），Ｗｉｌｅｙ，１９９６によれば、ワークステーション（ＳＰＡＲＣ２、米国サン・マイクロシステムズ社の商標））で、法数１０２４ビット・公開鍵８ビットのＲＳＡ暗号系を使って１０２４ビットのデータを処理する時間は、署名が０．９７秒、検証が０．０８秒かかっている。このため、スマートカードのような、演算能力の低いデバイスでは、認証の遅延を引き起こすという問題があった。

【００１４】

【課題を解決するための手段】本発明では、相互証明に用いられる認証方式としてゼロ知識証明方式を用いる。すなわち、本発明によれば、上述の課題を解決するために、証明者が有資格者であることを保証するセンターが生成する有資格情報を用い、証明者側の有資格性、および検証者側の認証の双方をゼロ知識証明方式を用いて行う相互認証方法において、証明者が不正な操作を施すことが困難な演算方法によって生成されたレスポンス生成情報に対して、センターが発行する有資格者情報を用いて該レスポンス生成情報に演算を施し、演算を施した情報をレスポンスとして検証者に提示することにより有資格者であることを証明するようにしている。

【００１５】このゼロ知識証明方式を用いた相互認証方式では、べき乗剰余計算に用いられる指数が数ビットから数十ビットで済むため、認証に必要な計算量を大幅に削減することができ、認証の遅延を防止することが可能となる。

【００１６】なお、本発明は、装置やコンピュータプログラム製品としても実現可能である。

【００１７】

【発明の実施の態様】以下では、認証者が検証者に送るコミットメントｗとして、

【００１８】

【数７】ｗ≡ｒ$^{v}$　ｍｏｄ　ｎを送り、検証者の認証をＲＳＡ暗号を説明する。ただし、上でｎは素因数分解が困難な二つの大きな素数ｐ，ｑの積からなる合成数法数、ｖはλ（ｐ－１，ｑ－１）と互いに素なｖ≧３を満たす数ビットから数十ビットの小さな素数とする（λ（ｘ，ｙ）はｘ，ｙの最小公倍数を表わす）。

【００１９】以下では、センタによって認証者に発行される有資格情報をアクセスチケットと呼ぶことにする。

【００２０】図１に本実施例の基本構成を示す。

【００２１】図１で、証明者１０１は認証プロトコルを実行する演算装置１０２、及び証明者の身許を保証するトークン１０３から成る。トークン１０３はスマートカードなどのタンパープルーフな（ｔａｍｐｅｒ－ｐｒｏｏｆ：内部のデータや処理を外部から観察困難な状態）装置で実現され、不正な方法では書き換えができない記憶装置１０５、及びレスポンス生成情報を生成する際に用いられ、検証者１１１との間で共有される一方向性ハッシュ関数ｈ、同じくレスポンス生成情報を生成する際に用いられ、センタ１２１との間で共有される秘密の一方向性ハッシュ関数Ｆを備えている。記憶装置１０５には証明者が受けたサービスに対応する課金情報などが記憶される。またトークン１０３には証明者１０１の有資格性を保証するためのアクセスチケット１０４が演算装置１０２を通して送信され、これを内部に保持することができる。

【００２２】検証者１１１は認証プロトコルを実行する演算装置１１２から成る。演算装置１１２は、チャレンジを生成する際に用いられ、認証者１０１と共有される

一方向性ハッシュ関数ｈを保持している。

【００２３】センタ１２１は、アクセスチケット生成装置１２３、証明者情報データベース１２４、アクセスチケット発行データベース１２５から成る。センタ１２１は証明者の要求に応じて、証明者情報データベース１２４、アクセスチケット発行データベース１２５から必要な情報を取り出し、アクセスチケット生成装置１２３においてアクセスチケット１２２を生成する。アクセスチケット生成装置１２３は、アクセスチケットを生成する際に用いられ、証明者１０１のトークン１０３との間で共有される秘密の一方向性ハッシュ関数Ｆを備えている。

【００２４】以下に証明者１０１、検証者１１１、センタ１２１が行う事前準備について説明する。

【００２５】センタ１２１は、証明者１０１の電話番号、ＦＡＸ番号、住所などから成る公開情報$I_A$を生成する。次にセンタ１２１は$I_A$にＩＳＯ９７９６などで定められる冗長性を加えて公開証明者情報$J_A$を生成し、これを公開する。

【００２６】またセンタ１２１は、この$J_A$から証明者秘密情報$C_A$を以下の式に従って生成する。

【００２７】

【数８】$C_A J_A^{-s} \mod n$ ここで、ｓはセンタ１２１のたす正の整数である。

【００２８】

【数９】$sv \equiv 1 \mod \lambda(p-1, q-1)$

公開指数ｖはλ（ｐ－１，ｑ－１）と互いに素な素数なので、上記を満たす数ｓが存在する。この値は、例えばユークリッドの互除法などで法λ（ｐ－１，ｑ－１）の下でのｖの逆数を求め、これをｓとすればよい。

【００２９】センタ１２１は、上式によって生成された$C_A$を証明者１０１に付帯させた状態で、証明者情報データベース１２４に格納する。

【００３０】次に検証者１１１は、ＲＳＡ暗号に基づいた認証を行うために、検証者公開鍵（$e_B$，ｎ’）、及び検証者秘密鍵$d_B$を生成する。ここで、ｎ’はＲＳＡ暗号に用いる法数で、素因数分解が困難な二つの大きな素数ｐ’，ｑ’の積からなる合成数法数とし、$e_B$はλ（ｐ’－１，ｑ’－１）と互いに素な小さな数とする。$e_B$を小さな数とする理由は検証者の認証を行う証明者１０１の計算量を少なくするためである。また検証者１１１の秘密鍵$d_B$を以下の式を満たすものとする。

【００３１】

【数１０】$e_B d_B \equiv 1 \mod \lambda(p'-1, q'-1)$

上記の式を満たす$d_B$を求めるには、例えばユークリッドの互除法などで法λ（ｐ’－１，ｑ’－１）の下での$e_B$の逆数を求め、これを$d_B$とすればよい。こうして生成した公開鍵（$e_B$，ｎ’）を公開し、$d_B$を検証者１１１の秘密情報として保持する。

【００３２】以上の検証者用の公開鍵、及び秘密鍵の生成はセンタがこれを行い、秘密裏に秘密鍵を検証者１１１へ送付してもよい。

【００３３】以下に証明者１０１の要求に応じて、センタ１２１がアクセスチケット１０４を生成する過程を説明する。

【００３４】まず、認証者１０１は検証者１１１から受けたいサービスをセンタ１２１に通知する。センタ１２１は証明者に対して、その証明者に付帯されたトークン秘密情報ｄ、及び証明者秘密情報$C_A$を証明者情報データベースから検索し、検証者１１１から受けるサービスからアクセスチケット識別子$I_t$、及び証明者１０１に割り当てられたサービス利用制限情報Ｌをアクセスチケット発行データベースから取り出す。サービス利用制限情報Ｌには、例えば年齢、資格などに応じた証明者の受けることのできるサービスの制限を記載する。

【００３５】センタ１２１はｄ，$C_A$，Ｌ，$I_t$から以下の式に従ってアクセスチケットｔを計算し、証明者１０１にインターネットなどを使って送付する。

【００３６】

【数１１】$t = C_A F(d, L, I_t)^{-1}$

次に認証の手順について説明する。認証の手順は以下のステップに従って実行される。

【００３７】［ステップ２００］証明者１０１は、トークン１０３において乱数ｒを生成する。

【００３８】［ステップ２０１］続いて証明者１０１は、ステップ２００で生成した乱数ｒを用いてコミットメントｗを以下の式に従って生成する。

【００３９】

【数１２】$w \equiv r^v \mod n$【００４０】［ステップ２０２］は、ステップ２０１で生成したコミットメントｗを演算装置１０２に送信する。

【００４１】［ステップ２０３］続いて証明者１０１は、演算装置１０２を用いて、ステップ２０１で生成したコミットメントｗ、検証者１１１から受けたいサービスに対応するアクセスチケット識別子$I_t$、及び対応するサービス利用制限情報Ｌを検証者１１１の演算装置１１２へ送信する。

【００４２】［ステップ２０４］検証者１１１は演算装置１１２において乱数ｒ’を生成する。

【００４３】［ステップ２０５］続いて検証者１１１は演算装置１１２においてステップ２０４で生成された乱数ｒ’、サービスに対応する課金情報等のメッセージμ、アクセスチケット識別子$I_t$を用いて、

【００４４】

【数１３】$\delta = \mu \mid I_t \mid r'$

を生成する。上記で「｜」はビット結合を表わす。

【００４５】［ステップ２０６］続いて検証者１１１は

演算装置１１２において、証明者１０１から送られてきたコミットメントwの下位ビットをδで置き換えることによって、w［δ］を生成する。

【００４６】［ステップ２０７］続いて検証者１１１は演算装置１１２においてステップ２０６で生成されたw［δ］を用いて、検証者１１１の認証データSを以下の式に従って生成する。

【００４７】

【数１４】$S \equiv w[\delta]^{dB} \mod n'$

【００４８】［ステップ２０８］続いて検証者１１１は演算装置１１２を用いて、ステップ２０７で生成された検証者１１１の認証データSと、課金情報などの証明者に対するメッセージμ、ステップ２０４で生成された乱数r'をチャレンジとして証明者１０１に送信する。

【００４９】［ステップ２０９］続いて検証者１１１は演算装置１１２を用いて、課金情報などの証明者に対するメッセージμ、ステップ２０４で生成された乱数r'、証明者１０１との間で共有している一方向性ハッシュ関数hを用いて、検証用指数gを以下の式に従って生成する。

【００５０】

【数１５】$g = h(r', \mu)$

【００５１】［ステップ２１０］認証者１０１は演算装置１０２を用いて、検証者１１１から送られてきた認証データSと、課金情報などの証明者に対するメッセージμ、チャレンジr'をトークン１０３へ送信する。

【００５２】［ステップ２１１］証明者１０１は、トークン１０３において、検証者１１１から送られてきた認証データS、及び検証者１１１の公開鍵$(e_B, n')$を用いて、以下の値S'を計算する。

【００５３】

【数１６】$S' \equiv S^{eB} \mod n'$

【００５４】［ステップ２１２］続いて証明者１０１は、トークン１０３において、ステップ２１１で生成したS'の上位ビットとステップ２０１で生成したwの上位ビットを比較し、同じ場合は検証者の認証が成功したものとしてステップ２１３へ進む。異なる場合は、不正な検証者と見なして認証の手順を中止する。

【００５５】［ステップ２１３］続いて証明者１０１は、トークン１０３において、検証者１１１から送られてきた乱数r'、課金情報などの証明者に対するメッセージμ、及び証明者１０１と共有している一方向性ハッシュ関数hを用いて、検証用指数g'を以下の式に従って生成する。

【００５６】

【数１７】$g' = h(r', \mu)$

【００５７】［ステップ２１４］続いて証明者１０１は、トークン１０３において、ステップ２００で生成した乱数r、ステップ２１３で生成した検証用指数g'、

$$w' \equiv J_A^{\ g} y^v \mod n$$

トークン１０３の記憶装置１０５内に記憶された証明者の秘密情報d、サービス利用制限情報L、アクセスチケット識別子$I_t$、センタ１２１と共有している秘密の一方向性ハッシュ関数Fを用いて、以下の式に従ってレスポンス生成情報y'を生成する。

【００５８】

【数１８】$y' \equiv rF(d, L, I_t)^{g'} \mod n$【００５９】は、ステップ２１３で生成した検証用指数g'、及びステップ２１４で生成したy'をトークン１０３から演算装置１０２へ送信する。

【００６０】［ステップ２１６］続いて証明者１０１は、検証者１１１から送られてきた課金情報などの証明者に対するメッセージμに基づき、トークン１０３内の記憶装置１０５内の情報を書き換える。

【００６１】［ステップ２１７］続いて証明者１０１は、演算装置１０２において、ステップ２１４で生成したy'、アクセスチケットt，及びステップ２１３で生成した検証用指数g'を用いて、以下の式に従ってレスポンスyを生成する。

【００６２】

【数１９】$y \equiv y' t^{g'} \mod n$【００６３】［ステップ２１は、ステップ２１７で生成したyを演算装置１０２から検証者１１１の演算装置１１２へ送信する。

【００６４】［ステップ２１９］検証者１１１は、演算装置１１２において、証明者１０１の公開証明者情報$J_A$、ステップ２０９で生成した検証用指数g、証明者１０１から送られてきたレスポンスy、及び公開指数vを用いて、以下の式に従って検証情報w'を生成する。

【００６５】

【数２０】$w' \equiv J_A^{\ g} y^v \mod n$【００６６】［ステップは、演算装置１１２において、証明者１０１から送られてきたコミットメントwとステップ２１９で生成した検証情報w'を比較し、二つが一致すれば証明者１０１の認証に成功したものとし、一致しなければ証明者１０１の認証に失敗したものとする。

【００６７】以下に、正当な証明者が正当なアクセスチケットを用いた場合には、検証者が計算するw'と証明者が生成したコミットメントwとが一致することを示す。

【００６８】証明者及び検証者上記のステップに従って計算をすると、明らかに

【００６９】

【数２１】$g = g'$

が成立するので、

【００７０】

【数２２】

$$\equiv J_A{}^{g} r^{v} F(d, L, I_t)^{gv} C_A{}^{gv} F(d, L, I_t)^{-gv}$$
$$\mod n \qquad \equiv J_A{}^{g} r^{v} C_A{}^{gv} \mod n \qquad \equiv r^{v} (J_A C_A{}^{v})^{g} \mod n$$

となる。ここで、ｓの条件より、

【００７１】

【数２３】$sv \equiv 1 \mod \lambda(p-1, q-1)$

が成り立っている。これにより、

【００７２】

【数２４】$sv = Q\lambda(p-1, q-1) + 1$

が成立する。ここで、上式でＱは適当な整数である。よって、

【００７３】

【数２５】
$$J_A{}^{-sv} = (J_A{}^{-1})^{Q\lambda(p-1, q-1)+1} \equiv J_A{}^{-1} (J_A{}^{-Q})^{\lambda(p-1, q-1)} \mod n$$

ここで、中国人剰余定理（Ｃｈｉｎｅｓｅ　Ｒｅｍａｉｎｄｅｒ　Ｔｈｅｏｒｅｍ）、及びオイラーの定理より、

【００７４】

【数２６】
$$(J_A{}^{-Q})^{\lambda(p-1, q-1)} \equiv 1 \mod n$$

が成り立つので

【００７５】

【数２７】$J_A{}^{-sv} \equiv J_A{}^{-1} \mod n$が成立する。よって

【００７６】

【数２８】
$$w' \equiv r^{v} (J_A J_A{}^{-sv})^{g} \mod n \equiv r^{v} (J_A J_A{}^{-1})^{g}$$

【００７７】上記の方法では、ゼロ知識証明方式を用いることにより証明者側の計算は数ビットから数十ビットの小さなべき指数に関するべき乗剰余計算を主体とした計算しか行わないため、大きなべき指数に関するべき乗剰余計算を行うＲＳＡ暗号を用いた方式に比較して、高速な認証が可能なことがわかる。

【００７８】上記では、証明者側の認証方式においてｖをλと互いに素なｖ≧３を満たす素数としたが、これをｖ＝２としても同じ効果が得られる。

【００７９】また、証明者側の認証方式、及び検証者側の認証方式として合成数ｎを法とする有理整数環の剰余類環の乗法群（Ｚ＝ｎＺ）*上で定義された演算を用いたゼロ知識証明、あるいはＳｃｈｎｏｒｒ署名方式を用いたゼロ知識証明方式、あるいは楕円曲線上で定義された有限群を用いたゼロ知識証明方式、あるいは素数ｐを法とする有理整数環の剰余類環の乗法群（Ｚ＝ｐＺ）*を用いたゼロ知識証明方式を用いても同様な効果が得られる。

【００８０】

【発明の効果】以上説明したように、ゼロ知識証明方式を用いた相互認証方式では、べき乗剰余計算に用いられる指数が数ビットから数十ビットで済むため、認証に必要な計算量を大幅に削減することができ、認証の遅延を防止することが可能となる。

【図面の簡単な説明】

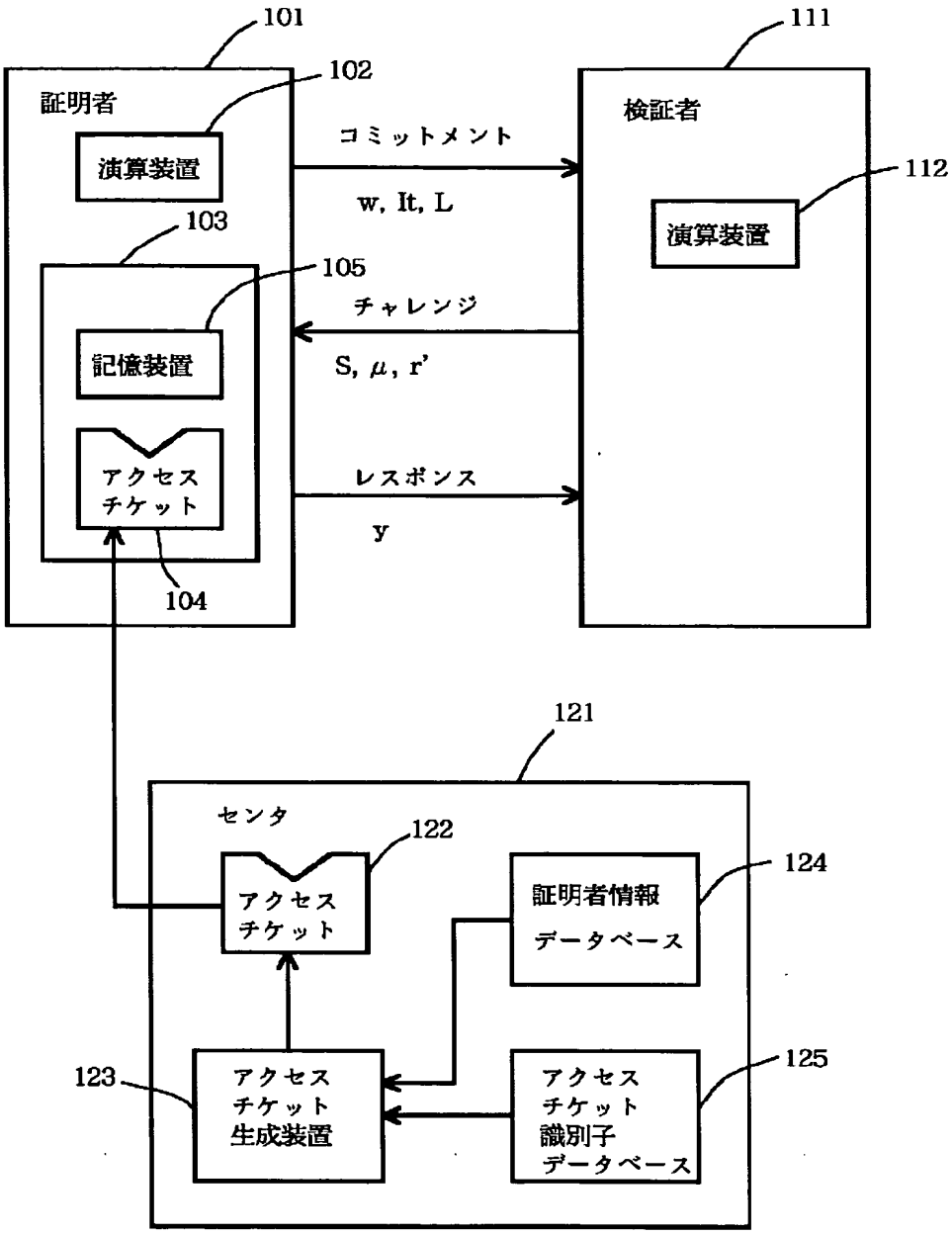
【図１】　本発明の実施例の構成を示すブロック図である。

【図２】　上述実施例の動作を説明するフローチャートである。

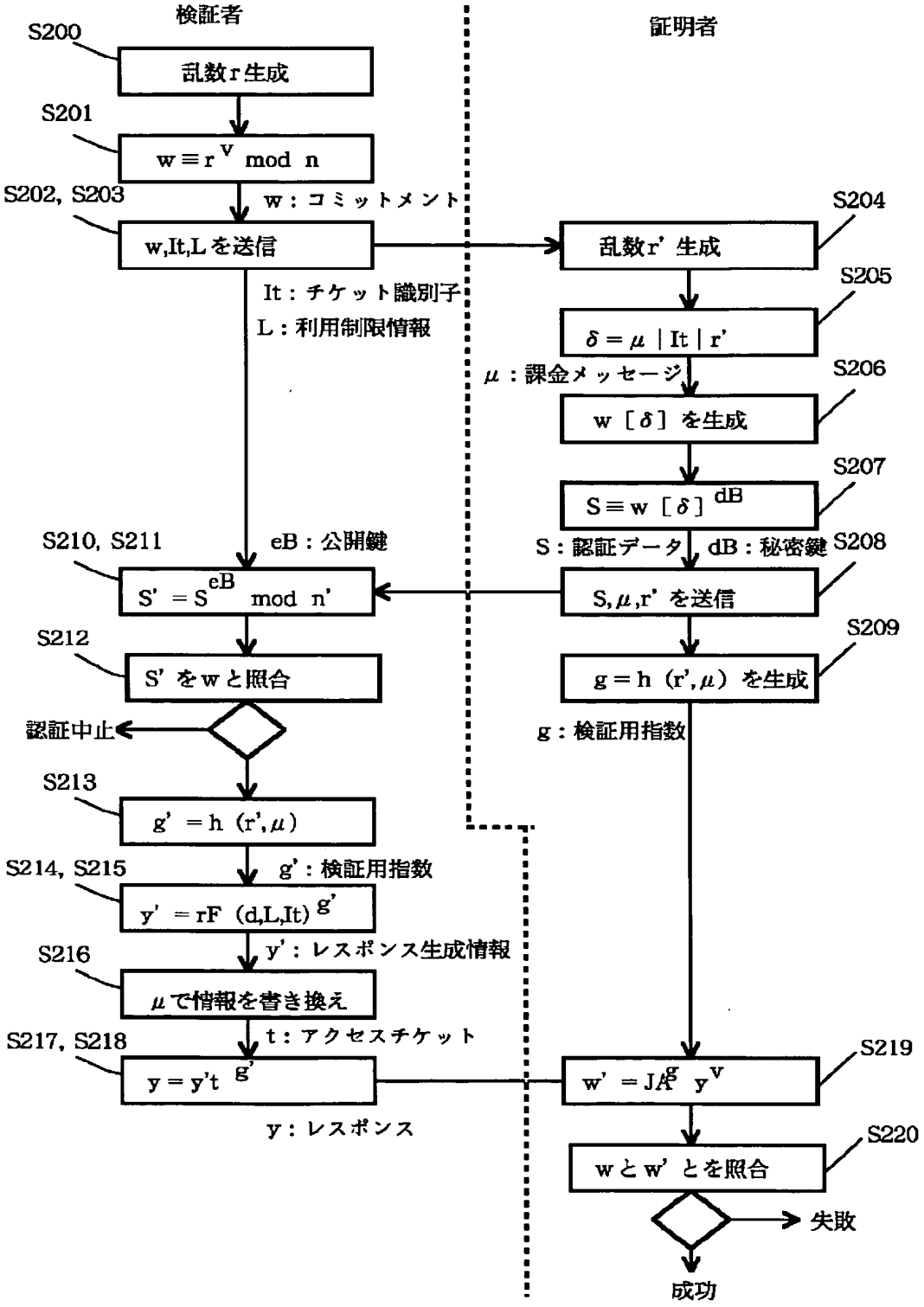
【符号の説明】

１０１　証明者$r^{v} \mod n$となり、認証が成功することがわ
１０２　演算装置
１０３　トークン１０４　アクセスチケット
１０５　記憶装置
１１１　検証者
１１２　演算装置
１２１　センタ
１２２　アクセスチケット
１２３　アクセスチケット生成装置
１２４　証明者情報データベース
１２５　アクセスチケット発行データベース

【図１】


101
111
証明者
102
検証者
演算装置
コミットメント
w, It, L
112
演算装置
103
105
チャレンジ
記憶装置
S, μ, r'
アクセス
チケット
レスポンス
y
104
121
センタ
122
アクセス
チケット
証明者情報
データベース
124
123
アクセス
チケット
生成装置
アクセス
チケット
識別子
データベース
125

【図２】


検証者
証明者
S200
乱数r生成
S201
w≡r^v mod n
S202, S203
w：コミットメント
w,It,Lを送信
It：チケット識別子
L：利用制限情報
S204
乱数r' 生成
S205
δ＝μ｜It｜r'
μ：課金メッセージ
S206
w［δ］を生成
S207
S≡w［δ］^dB
S：認証データ
dB：秘密鍵
S208
S,μ,r' を送信
S209
g＝h（r',μ）を生成
g：検証用指数
S210, S211
eB：公開鍵
S'＝S^eB mod n'
S212
S' をwと照合
認証中止
S213
g'＝h（r',μ）
S214, S215
g'：検証用指数
y'＝rF（d,L,It）^g'
S216
y'：レスポンス生成情報
μで情報を書き換え
S217, S218
t：アクセスチケット
y＝y't^g'
y：レスポンス
S219
w'＝JA^g y^v
S220
wとw'とを照合
失敗
成功